VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Šachový klient

Chess client

2010

David Juráň

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 3. května 2010 .....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla.

Abstrakt

Cílem této diplomové práce je vytvořit šachového klienta pro server FICS. V práci se nejdříve zaměřuji na zmapování služeb, poskytovaných serverem FICS. V druhé části se snažím o návrh aplikace pomocí architektury MVC pro snažší použitelnost kódu pro více grafických rozhraní. Následuje prozkoumání technologií Eclipse RCP a Eclipse RAP včetně implementace klienta pomocí těchto technologií. Dále vytvářím databázi pro ukládání tahů partií pro případnou pomoc uživateli při zahájení hry.

Klíčová slova: šachy, FICS, Eclipse RAP , Eclipse RCP

Abstract

The aim of this diploma thesis is to create the chess client for Free Internet Chess Server (FICS). At the very beginning I concentrate on charting the already existing servises provided by FICS. In the second part I try to draw up an application by means of the architecture MVC, which facilitates the code usage in more graphical user interfaces. Afterwards I analyse both the Eclipse RCP and Eclipse RAP technologies. These technologies are then used for implementation of client. In the next part of the thesis I have created the experience base for saving moves of a game, which can be useful for a user when opening a game.

Keywords: chess, FICS, Eclipse RAP , Eclipse RCP

Seznam použitých zkratek a symbolů

API	–	Application Programming Interface
CSS	–	Cascading Style Sheets
FICS	–	Free Internet Chess Server
GPL	–	General Public License
GUI	–	Graphical User Interface
ICC	–	Internet Chess Club
ICS	–	Internet Chess Server
IDE	–	Integrated Development Environment
JAR	–	The Java Archive Tool
MVC	–	Model-view-controller
NASA	–	National Aeronautics and Space Administration
OSGi	–	Open Services Gateway initiative
PGN	–	Portable Game Notation
PHP	–	PHP: Hypertext Preprocessor
RAP	–	Rich AJAX Platform
RCP	–	Rich Client Platform
SQL	–	Structured Query Language
SŘBD	–	Systém Řízení Báze Dat
SWT	–	Standart Widget Toolkit

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

1.1 Motivace

Dlouhou dobu hraji rekreačně šachy. Postupem času se mi v tomto ohledu stal výborným pomocníkem počítač. Nejdříve jsem jej využíval ke hře proti počítači a později jsem objevil šachový server FICS. Tento server mi umožnil spojit se s hráči z celého světa a hrát s nimi šachy. Bohužel s přechodem na operační systém Linux jsem již nadále nemohl využívat stávajícího klienta (BabasChess), ale musel jsem si vybrat z nabídky klientů pro Linux. Rozhraní těchto klientů mi však nevyhovovalo. Proto jsem se rozhodl naimplementovat vlastní řešení šachového klienta pro FICS.

1.2 Cíle práce

Cílem této práce je naiplementovat šachového klienta pro server FICS. Aby tento klient mohl pokrýt co největší množství uživatelů serveru, bude klient navrhnut jako multiplatformní včetně webového klienta. Součástí desktopové aplikace bude znalostní báze, která se bude snažit uživateli pomoci při zahajovací fázi partie a to prostřednictvím tahů, které získal buď z odehraných tahů uživatele nebo z tahů uživatelů serveru. Znalostní báze si neklade za cíl pokrýt celou partii, protože to vzhledem k počtu možností v pokročilejší části hry ani není možné.

2 Služby šachových serverů

Následující kapitola se zabývá možnostmi šachových serverů a jejich komunikačními protokoly.

2.1 Dostupné servery

Na internetu se v dnešní době vyskytuje velké množství šachových serverů, které poskytují možnost hraní této hry online proti lidským soupeřům. Tyto servery se dají rozdělit na dva typy:

- poskytují pouze vlastní webové rozhraní
- poskytují navíc také přístup k serveru bez nutnosti použití jejich webového rozhraní

Serverů druhého typu je menšina. I tato skupina se dá rozdělit na dvě podskupiny a to placené a neplacené. Zaměřím se na neplacenou část. Zde se mi podařilo najít pouze dva zástupce:

Internet Chess Club (www.chessclub.com) - server poskytující rozhraní pro připojení šachových klientů. Nevýhodou je, že možnost bezplatného hraní na tomto serveru je omezena pouze na 7 dní, a to jen s klientem, poskytovaným tímto serverem.

Free Internet Chess Server (www.freechess.org) - jedná se o server, který umožňuje bezplatné hraní šachu za pomocí poskytovaných klientů nebo pomocí appletu aplikace Jin, umístěné na domovských stránkách.

2.2 Služby šachového serveru FICS a jejich komunikační protokoly

2.2.1 Proměnné serveru

Proměnné slouží k nastavení chování serveru což je výhodné pro specifické nastavení pro jednotlivé šachové klienty. Tyto proměnné se dělí na dva typy, a to proměnné uživatelské a proměnné rozhraní. Těmto typům a jejich podrobným popisům se věnují následující dvě podkapitoly.

2.2.1.1 Uživatelské proměnné Tyto proměnné jsou určeny pro upravení vlastností serveru důležitých pro uživatele. Jsou vždy ukládány v rámci profilu hráče, takže jsou nastaveny na požadované hodnoty při každém přihlášení. V případě hráče používajícího guest účet jsou nastaveny serverem na optimální hodnoty pro hraní.

Použití:

nastavení proměnné: set <název proměnné> [hodnota]

seznam všech uživatelských proměnných: variables [login hráče]

Seznam vybraných proměnných:

v_chanoff umožní vypnout naslouchání veškerých komunikací v rámci serveru.

v_flip nastavuje orientaci šachovnice.

v_gin upozorňuje na hry, které na serveru začínají nebo končí.

v_time a v_inc standardní nastavení času a přírůstku pro šachovou partii, která je žádaná bez parametrů.

v_interface umožní nastavení jména klienta. Tato informace není ukládána, takže se po odhlášení nastaví na výchozí hodnotu.

v_seek upozorňuje na nabízené hry ostatními hráči.

v_style proměnná nastavuje, v jakém formátu budou klientu ze serveru posílány informace o aktuální pozici na šachovnici.

2.2.1.2 Proměnné rozhraní Tyto proměnné se používají k upravení výstupu serveru pro umožnění parsování informací pomocí klienta. Hodnoty se neukládají při odhlášení, a proto si je musí při každém přihlášení klient znovu nastavit. Uživateli se nedoporučuje tyto hodnoty měnit, protože by mohlo dojít ke špatné funkčnosti programu.

Použití:

nastavení proměnné: iset <název proměnné> [hodnota]

seznam všech proměnných klienta: ivariables [login hráče]

Seznam vybraných proměnných:

seekinfo posílá informace o nabízených hrách.

compressmove posílá pouze zjednodušené informace o tazích.

gameinfo posílá parametry hry (typ hry, čas, ...) při každém vytvoření nové hry nebo pozorování cizí hry.

pendinfo zlepšený zápis informací o nabídnutí hry.

lock uzamkne možnost změny proměnných do příštího přihlášení. Tato proměnná zabraňuje uživateli měnit nastavení a přivodit tak nefunkčnost programu.

2.2.2 Nabízené hry

Hru lze na serveru vytvořit pomocí nabízení her. Hru lze nabídnout buď určitému hráči nebo všem (popřípadě lze nastavit některá kritéria, jako například rating hráčů, kterých se tato nabídka týká). Pokud hráči mají zapnutou funkci sledování nabízených her, tak vidí všechny nabídky, ze kterých si mohou vybrat a hru přijmout.

Protokol:

Poslání výzvy hráči: match [user] [rated|unrated] [čas] [přírůstek] [White|Black]

Poslání výzvy všem: seek [čas] [přírůstek] [rated|unrated] [white|black] [crazyhouse] [suicide] [wild #] [auto|manual] [formula] [rating-range]

- čas - klasická délka hry (v minutách)
- přírůstek - k času hráče se při každém jeho tahu přičte hodnota daná přírůstkem (v sekundách)
- rated|unrated - jestli se bude výsledek započítávat do hodnocení hráče
- white|black - bílé nebo černé figury
- crazyhouse - styl hry, při kterém při sebrání figury ji získáváte na svou stranu
- suicide - styl hry, při které je cílem přijít o všechny své figury
- wild - další styl hry, který má více typů, proto následuje číslo
- auto|manual - zda má hra začít automaticky v okamžiku, kdy soupeř přijme výzvu nebo máte být dotázáni
- formula - jestli se má použít formule pro filtrování odpovědí na výzvu
- rating-range - lze zvolit Rating soupeřů, pro které je výzva určena (například 900-1500)

Zobrazení výzvy pro určitého hráče serverem: Challenge: (hráč1) (hráč2) [rated|unrated] [typ hry][čas] [přírůstek]. You can "accept" or "decline", or propose different parameters.

Zobrazení obecné výzvy serverem(při zapnuté proměnné "seekinfo"): <s> [číslo hry] w=[login] ti=02 rt=[rating] t=[čas] i=[přírůstek] r=[rated|unrated] tp=[typ] c=[barva] rr=[rozsah ratingu] a=[automatické přijetí] f=[formula]

2.2.3 Statistiky

Server nabízí pár možností jak se informovat o statistikách. Jednak je to zjištění informací o posledních deseti hrách určitého hráče, anebo počet všech her (výher, proher, remíz), rozdělený do kategorií podle typu her.

2.2.3.1 Protokol:

Posledních deset her: history <login hráče>

Celková statistika her: stats <login hráče>

Další statistiky a analýzy lze nalézt na adrese www.ficsgames.com, kde se nachází velká databáze her odehraných na serveru. Zde lze pomocí parametrů najít všechny vaše odehrané hry a také hry soupeřů. V současné době je v této databázi uloženo přes 110 000 000 odehraných her.

2.2.4 Hra

Hra je základní část, kterou server poskytuje. Pro ovládání hry se používají tahy, ale také příkazy, které umožňují řízení hry.

2.2.4.1 Podporované typy her: Rozdělení základních typů her (lightning, blitz, standard) je rozděleno podle "předpokládané délky". "Předpokládaná délka" vychází z údaje, že hra obsahuje 40 tahů. Předpokládaná doba hry se vypočítá vztahem:

```
Předpokládaná doba = (Počáteční čas + Přírůstek * 2 / 3)
```

lightning - jedná se o tento typ hry v případě, že "předpokládaná doba" partie je kratší než 3 minuty pro jednoho hráče.

blitz - hra patří do typu blitz v případě, že "předpokládaná doba" hry se pohybuje mezi třemi a patnácti minutami pro každého hráče.

standard - jedná se o tento typ hry v případě, že "předpokládaná doba" partie je delší než 15 minut pro jednoho hráče.

nonstandard - jde o hru, kde má každý hráč jinou délku času pro hraní partie.

Suicide chess - jedná se o speciální druh hry kdy není cílem mat soupeře, ale hlavním cílem je ztratit všechny figury (včetně krále). Ve hře platí speciální pravidla oproti klasickému šachu jako:

- v případě možnosti sebrání figury protihráče musí být tento tah proveden.
- rošáda není povolena.
- král je brán jako ostatní figury, tedy není možné dát šach nebo šachmat.

losers - cílem hry je přijít o všechny figury až na krále. Rozdíl proti klasickému šachu je, že v případě možnosti sebrání figury protihráče musí být tento tah proveden. Výhry může být dosaženo následujícími způsoby:

- ztráta všech figur kromě krále.
- král se ocitl ve věčném šachu
- šachmat soupeřem.

untimed - jedná se o hru, která není časově omezena. Není možno hrát hodnocenou partii.

wild - tato hra má osm podvariant, které se liší ve většině případů v počátečním rozložení figur. (Tato varianta není aplikací podporována)

atomic - varianta stejná jako klasické šachy s tím rozdílem, že pěšec exploduje při sebrání soupeřovy figury.

crazyhouse - rozdílnost oproti klasickému šachu je v tom, že v případě sebrání soupeřovy figury uživatel dostane stejnou figuru ve své barvě. Tato figura může být během partie vložena na kterékoliv volné pole (výjimku tvoří pěšec, který nesmí být vložen do 1 nebo 8 řady šachovnice). (Tato varianta není aplikací podporována)

2.2.4.2 Příkazy ve hře:

- abort - požadavek, aby byla hra zrušena
- adjourn - požadavek, aby byla hra odložena na jindy
- draw - požadavek pro remízu
- flag - slouží k ukončení partie, když protihráči vypršel čas a server to nezaznamenal
- moves - server pošle tahy ke hře, kterou hrajete
- pause - požadavek o krátké pozdržení hry
- promote - nastavuje, na jakou figuru se má změnit pěšec, když dosáhne posledního pole
- refresh - pošle znovu pozici všech figur na šachovnici
- resign - tímto příkazem hráč vzdává partii
- switch - požadavek na výměnu barev figur
- takeback - požadavek na vrácení jednoho tahu
- unpause - požadavek o znovuspuštění pozdržené hry
- withdraw - zrušení požadavku, který jste udělal

Nejdůležitější částí pro hru jsou ale tahy. Jedná se o informace, které jsou posílány serveru, když provedete tah a server je dále pošle vašemu protihráči. Stejně tak, když váš soupeř provede tah, jeho klient pošle informace o tahu serveru a server následně vám. FICS nabízí 13 možností, jak tyto informace předávat. Jako výchozí je na serveru používán styl "Standard ICS board in ascii format", který není vhodný pro zpracování počítačovým programem, ale je určen pro zobrazení partií lidem, kteří nepoužívají šachové programy pro zobrazování partií. Ale mezi nejpoužívanější patří "style12", který také používám ve svém klientu. Jedná se o styl, který je kompatibilní se stylem používaným serverem ICC.

2.2.4.3 Protokol:

vámi provedený tah: [pozice odkud][pozice kam] například e2e4

2.2.4.3.1 Styl Standard ICS board in ascii format

```
Game 15 (GuestHFNW vs. GuestGBYJ)

    ---------------------------------
 8  | *R| *N| *B| *Q| *K|   |   | *R|     Move # : 5 (White)
    |---+---+---+---+---+---+---+---|
 7  | *P| *P|   | *P| *P| *P| *B| *P|     Black Moves : 'Bg7'
    |---+---+---+---+---+---+---+---|
 6  |   |   |   |   |   | *N| *P|   |
    |---+---+---+---+---+---+---+---|
 5  |   |   | *P| P |   |   |   |   |     Black Clock : 5:06
    |---+---+---+---+---+---+---+---|
 4  |   |   | P |   |   |   |   |   |     White Clock : 5:00
    |---+---+---+---+---+---+---+---|
 3  |   |   | N |   |   |   |   |   |     Black Strength : 39
    |---+---+---+---+---+---+---+---|
 2  | P | P |   |   | P | P | P | P |     White Strength : 39
    |---+---+---+---+---+---+---+---|
 1  | R |   | B | Q | K | B | N | R |
    ---------------------------------
      a   b   c   d   e   f   g   h
```

- Game 15 - číslo probíhající hry
- (GuestHFNW vs. GuestGBYJ) - Jména hráčů
- *R - jedná se o černou figuru
- R - jedná se o bílou figuru
- Move # - číslo tahu a barva hráče na tahu
- Moves - poslední tah
- Black Clock - čas zbývající černému hráči
- White Clock - čas zbývající bílému hráči
- Black Strength - síla černých figur
- White Strength - síla bílých figur

2.2.4.3.2 Styl style12

```
 <12> rnbqkb-r pppppppp -----n-- -------- ----P--- --------
PPPPKPPP RNBQ-BNR B -1 0 0 1 1 0 7 Newton Einstein 1 2 12 39 39
119 122 2 K/e1-e2 (0:06) Ke2 0
```

- "<12>" identifikace, že se jedná o tah popsaný protokolem style12
- 8 polí, která reprezentují jednotlivá pole na šachovnici
- barva figur toho, který je na tahu ("B" nebo "W")
- -1 jestliže tah nebyl o dva pěšcem, jinak pozice, kde byl tah o 2 proveden
- může bílý stále udělat krátkou rošádu (0=ne, 1=ano)
- může bílý stále udělat dlouhou rošádu
- může černý stále udělat krátkou rošádu
- může černý stále udělat dlouhou rošádu
- počet tahů. Jestliže je >= 100, hra může být považována za remízu
- číslo hry
- login hráče s bílými figurami
- login hráče s černými figurami
- vaše pozice v této hře:

  -3 při zobrazování pouze pozic
  -2 pozorujete hru, která je hodnocena
  2 jste hodnotící této hry
  -1 hrajete tah protihráče
  1 hrajete váš tah
  0 pozorujete hru hranou jinými hráči

- čas partie (v sekundách)
- přírůstek (v sekundách)
- Materiální hodnota figur bílého
- Materiální hodnota figur černého
- Zbývající čas bílého
- Zbývající čas černého

- číslo tahu (standardní počítání: první tah bílého i černého počítán jako 1)
- speciální zápis předchozího tahu (”none”, jestliže žádný nebyl)
- čas, který trval předchozí tah ”(min:sec)”.
- zjednodušený zápis předchozího tahu (”none”, jestliže žádný nebyl)
- orientace šachovnice: 1 = černé figury dole na šachovnici, 0 = bílé figury dole na šachovnici

[5]

2.2.5 Odložené hry

Funkce odložených her slouží tehdy, když nějaká hra nemůže být dohrána. Stává se tak, když o odložení hry požádá jeden z hráčů (druhý hráč může nebo nemusí žádost přijmout), nebo když jeden z hráčů ztratí spojení se serverem. V tento moment se pozice hry ukládá na server až do té doby, než ji hráči dohrají. Každý hráč může mít však maximálně 16 odložených her. Další pravidlo je, že pokud chceme hrát hru s hráčem, se kterým již máme odloženou hru, musíme nejdříve dohrát tu odloženou.

2.2.5.1 Příkazy:

požadavek o odložení hry: adjourn

obnovení odložené hry: match|resume

zobrazení seznamu vašich odložených her: stored

zobrazení pozic figur v odložené hře: sposition <login hráče>

přehodnocení adminem při nemožnosti hru dohrát: message adjudicate <login bílého> <login černého> <výsledek> <poznámky>

2.2.6 Chat

Server poskytuje službu, která umožňuje komunikaci mezi hráči. Komunikace se dá na serveru rozdělit do tří částí: Komunikace při hře, Zprávy určitému hráči, Komunikace v místnostech.

2.2.6.1 Komunikace při hře: Jedná se o zprávy mezi lidmi, kteří buď hrají, anebo pozorují probíhající hru.

Protokol:

Poslání zprávy: say <text zprávy>

Zobrazení zprávy serverem: game [číslo hry] <text zprávy>

2.2.6.2 Zprávy určitému hráči: Jde o poslání zprávy určitému hráči. Existují dva druhy a to přímé poslání zprávy a poslání zprávy, která bude uložena na server (výhoda toho je, že může být poslána zpráva i hráči, který není zrovna připojen).

Protokol:

Poslání zprávy: tell [login hráče] <text zprávy>

Zobrazení zprávy serverem: [login hráče] tells you <text zprávy>

Zpráva ukládána na server: messages [login hráče] <text zprávy>

Přečtení zprávy ze serveru: messages [číslo zprávy | login hráče]

2.2.6.3 Komunikace v místnostech: Server poskytuje ke komunikaci také okolo 250 místností, do kterých je možné se přihlásit a diskutovat o určitých tématech. Témata jsou dána názvy místností.

Protokol:

Poslání zprávy: tell [číslo místnosti] <text zprávy>

Zobrazení zprávy serverem: [login hráče] (číslo místnosti) <text zprávy>

3 Popis Eclipse RCP a RAP technologií

Eclipse je open source komunita, která vyvíjí projekty jako jsou frameworky, nástroje a prostředí pro vývoj softwaru. Projekt Eclipse byl vytvořen firmou IBM v roce 2001. Eclipse RAP a Eclipse RCP jsou jedny z mnoha projektů, spadající pod projekt Eclipse.

3.1 Eclipse RCP

Eclipse Rich Client Platform je platforma, která byla vyvinuta pro urychlení vývoje desktopových aplikací. Jedná se o skupinu pluginů vhodných pro vývoj Rich Client aplikace. V případě Eclipse to znamená, že pro vývoj jakékoliv RCP aplikace potřebujeme pluginy org.eclipse.ui a org.eclipse.runtime. Eclipse Rich Client platforma obsahuje tyto komponenty:

- Jádro platformy
- Equinox OSGi
- Standard Widget Toolkit (SWT)
- Eclipse Workbench
- JFace

3.1.1 Historie

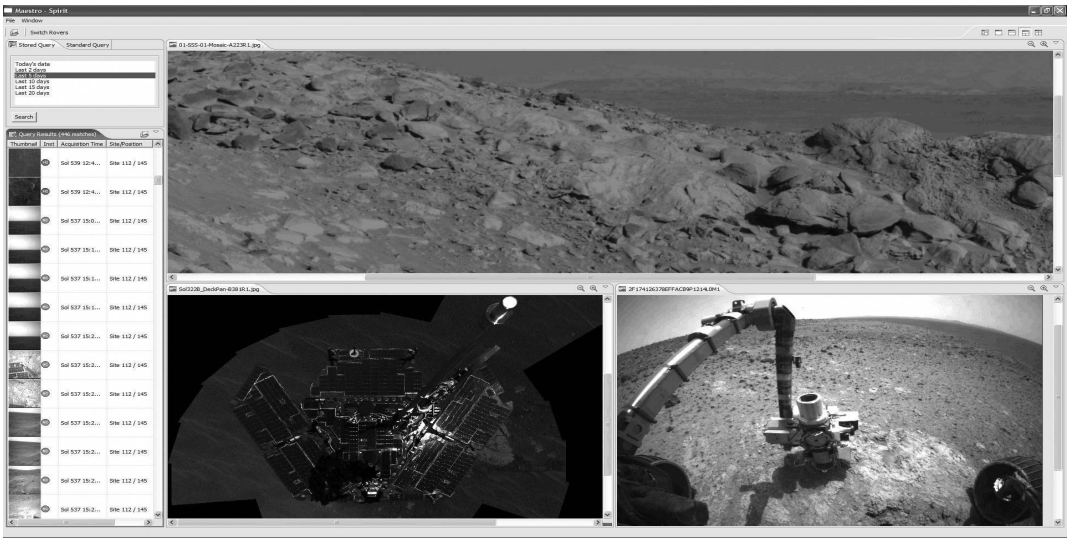
Cílem projektu Eclipse nebylo ze začátku vytvořit platformu pro vývoj RCP aplikací, ale vytvořit platformu pro integrování vývojářských nástrojů. První iniciativa vznikla ve verzi Eclipse 2.1 (vydáno 27. března 2003) a to proto, že IDE vyvinuté s použitím Eclipse mělo dobrý výkon a dobře vypadalo. Eclipse jako Rich Client platforma vyšlo ve verzi 3.0 (vydáno 28. června 2004). V této verzi byly eliminovány všechny závislosti, které byly obsaženy v rámci Eclipse IDE. Dále bylo otevřeno uživatelské rozhraní pro přizpůsobení. Základem pro dynamické přidávání a odstraňování pluginů se stalo runtime, založené na OSGI. [1]

3.1.2 Ukázky použití

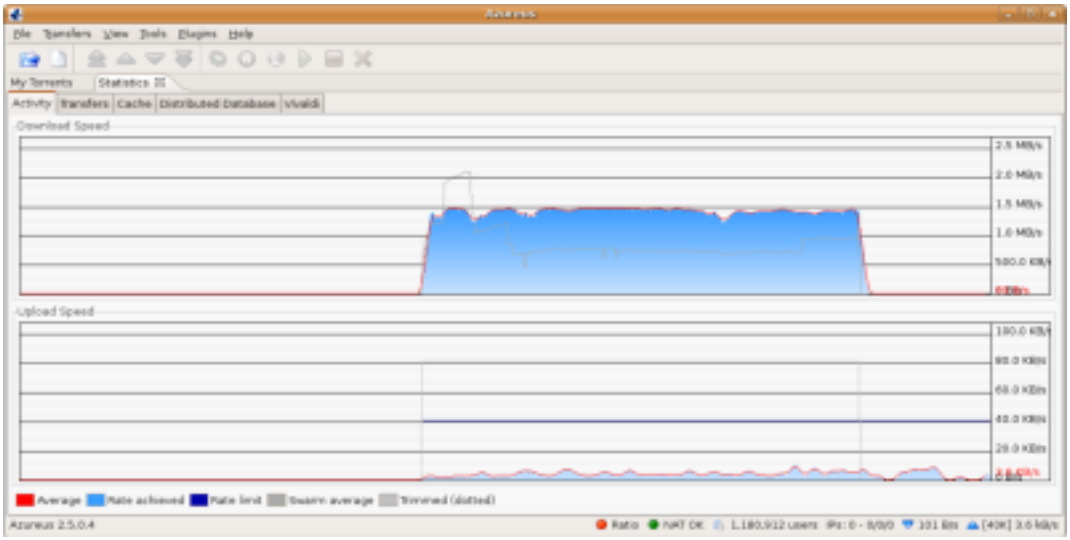
Eclipse začalo využívat mnoho firem pro vývoj vlastních desktopových aplikací. Následují ukázky některých z těchto aplikací.

3.1.2.1 Maestro Aplikace vyvinutá organizací NASA, která je využívána při některých z vesmírných misí. Ukázka na obrázku číslo 1.

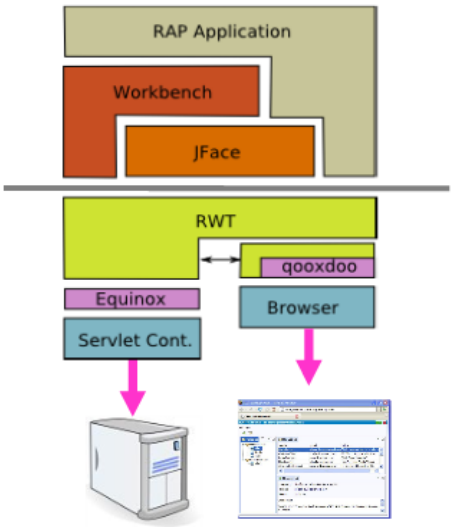
3.1.2.2 Azureus Vznikl v roce 2003 spíše pro experimentování s grafickou knihovnou SWT z Eclipse. Jedná se o populární BitTorrent klient, který je dostupný pro většinu používaných operačních systémů. Ukázka na obrázku číslo 2.



Obrázek 1: Ukázka aplikace Maestro [1]



Obrázek 2: Ukázka aplikace Azureus



Obrázek 3: Architektura Eclipse RAP [10]

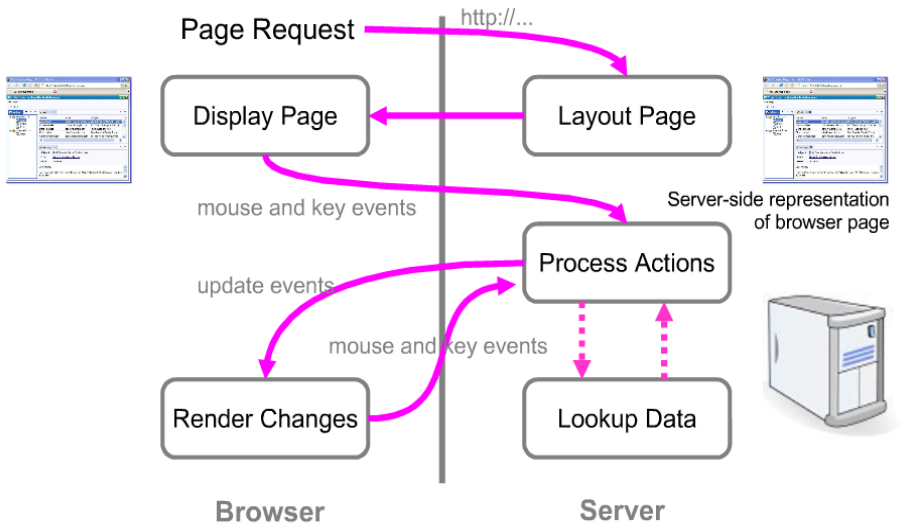
3.2 Eclipse RAP

Eclipse Rich Ajax Platform (RAP) umožňuje vývojářům vytvářet webové aplikace, využívající AJAX s použitím jazyka JAVA. Technologie RAP se snaží o to, aby bylo možné použít co největší množství kódu z aplikací, vytvořených pomocí Eclipse RCP. RAP je vyvinut tak, aby pracoval na všech populárních webových prohlížečích. RAP je velmi podobný projektu Eclipse RCP. Na rozdíl od něj neběží jako desktopová aplikace, ale běží na serveru a můžeme k ní přistupovat přes klasické webové prohlížeče. Toho je dosaženo pomocí speciální implementace SWT (RWT). Aplikace jsou vyvíjeny jako pluginy za pomocí Javy a exportují se do .war souborů, díky nimž mohou běžet i mimo Eclipse IDE.

3.2.1 Architektura

Architektura je zobrazena na obrázku 3. Popis jednotlivých částí:

- Workbench - poskytuje komunikační infrastrukturu s komponentami. Dále umožňuje hierarchické a logické uspořádání komponent uživatelského rozhraní.
- JFace - jedná se o rozšíření k SWT. Poskytuje přátelštější API pro vývojáře, kterým nedostačuje API SWT. Nabízí více předpřipravených komponent, například dialogy, akce a další.
- RWT - SWT bylo přeimplementováno pro použití v RAP aplikacích na RAP Widget Toolkit (RWT). Místo kreslení na `GraphicalContext` objekty komunikují s bufferem obrazovky operačního systému a RWT vytváří potřebné JavaScriptové příkazy, díky kterým qooxdoo vykresluje příslušné widgety.



Obrázek 4: Komunikace Eclipse RAP [10]

- qooxdoo - open source JavaScriptový framework. Poskytuje základní JavaScriptový framework pro různé prohlížeče a AJAX. Navíc nabízí GUI knihovnu, která umožňuje vytváření webových aplikací bez znalostí HTML nebo CSS.
- Equinox - umožňuje vývojářům vytvářet aplikace pomocí balíků. Jedná se o implementaci specifikace OSGi R4 core frameworku.

Komunikace mezi prohlížečem a serverem v rámci Eclipse RAP je znázorněna na obrázku číslo 4.

3.2.2 Historie

Projekt RAP vznikl v březnu roku 2006 a projektem Eclipse se stal v červnu 2006. Plánem bylo vytvořit implementaci prostředí s použitím W4T toolkitu. Z důvodu, že Eclipse komunita upřednostňuje SWT toolkit, se vývojáři rozhodli vytvořit RWT, což je podmnožina SWT toolkitu, který je využíván u Eclipse RPC technologie. Verze 1.0 vyšla 10. října 2007.[2]

3.2.3 Vzhled

Protože vzhled desktopových aplikací není v prostředí webu vždy žádoucí, byla vyvinuta možnost měnit vzhled RAP aplikací. Toto je umožněno pomocí CSS verze 2.1. Následuje ukázka (výpis číslo 1) jak lze měnit vzhled RAP aplikace:

```
* {
    backgound-color: white;
  }
```

```
Button[PUSH], Button[TOGGLE] {
  border: 2px solid blue;
  color: rgb( 17, 23, 103 );
  background-color: #f9f9f9;
}

Button[PUSH]:hover, Button[TOGGLE]:hover {
  background-color: white;
}
```

Výpis 1: Nastavování vzhledu pomocí CSS v Eclipse RAP [7]

Komunita Eclipse také vyvíjí témata, která se používají ke změně celého vzhledu aplikace. Jsou prozatím k dispozici pouze dvě, a to témata business a fancy (stále ve vývoji).

3.2.4 Podporované prohlížeče

- Microsoft Internet Explorer 5.5, 6 a 7
- Firefox 1.0 a novější
- Opera 8 a novější
- Safari 3.0
- všechny prohlížeče s jádrem Gecko 1.7 a vyšším

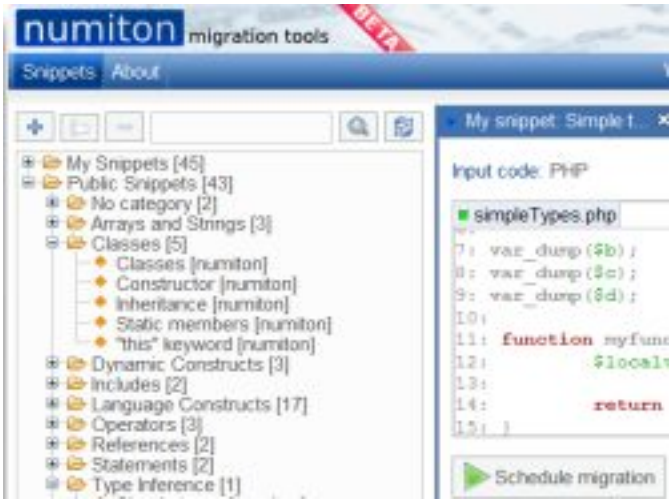
3.2.5 Ukázky použití

3.2.5.1 Numiton Migration Tools Tato aplikace umožňuje automatický převod softwaru z jednoho jazyku do druhého. V aktuální verzi je umožněn převod kódu z PHP do jazyku JAVA. Ukázka na obrázku číslo 5.

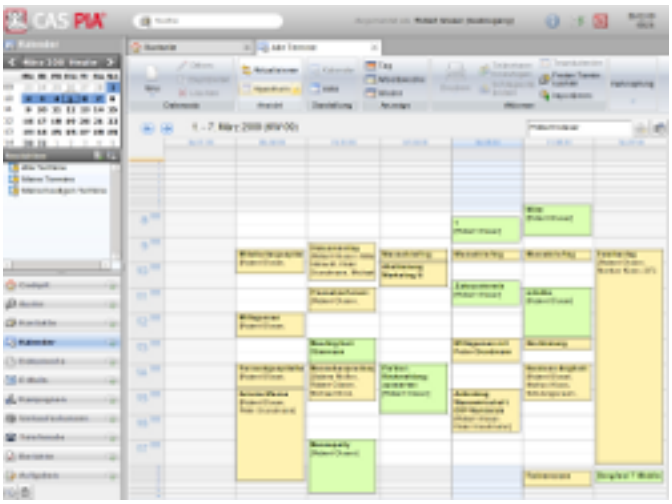
3.2.5.2 CAS PIA Jedná se o aplikaci, která nabízí funkce jako je správa kontaktů, kalendář a další. Ukázka na obrázku číslo 6.

3.3 Single sourcing

Jedna z velkých výhod technologií RAP a RCP je možnost použití takzvaného single sourcingu. Jedná se o koncept umístění aplikace na více platforem s použitím stejného kódového základu. Vychází se z myšlenky "write once, run everywhere". Volný překlad je napiš jednou a spusť všude. Java sama o sobě již tuto myšlenku splňuje tím, že je možno spustit aplikaci na mnoha operačních systémech. RAP v tomto ohledu zachází ještě dále, když umožňuje použití kódu RCP při vytváření webové aplikace. [2]



Obrázek 5: Ukázka aplikace Numition Migration Tools [8]



Obrázek 6: Ukázka aplikace CAS PIA [8]

3.3.1 Víceuživatelský přístup

Zatímco Eclipse RCP umožňuje přístup k aplikaci jenom jednomu uživateli , Eclipse RAP umožňuje víceuživatelský přístup. To znamená větší spotřebu paměti, ale také to, že k objektům, které byly ve Workbench Eclipse RCP vyvinuty jako singleton, se musí jinak přistupovat. Je tomu tak z důvodu, že každý uživatel má jiná specifická data v rámci programu. RAP toto řeší pomocí `SessionSingletons`, které jsou jedinečné pro každého uživatele. Implementace vzoru singleton v RAP je ukázána ve výpisu číslo 2. [2]

```
public class MySingleton extends SessionSingletonBase {

  private MySingleton(){
    // soukromy konstruktor
  }

  public static MySingleton getInstance() {
    return (MySingleton)getInstance(MySingleton.class);
  }
  // zbytek kodu
}
```

Výpis 2: Implementace singletonu v Eclipse RAP [9]

Dále je možno k singletonu přistupovat pouze z vláken, která jsou asociovány se session uživatelského rozhraní. Ze všech "non-UI" vláken se přístup k singletonu trochu liší. Tento přístup lze vidět ve výpisu číslo 3.

```
final Display display = Display.getCurrent();
final Runnable runnable = new Runnable(){
  public void run() {
    UICallBack.runNonUIThreadWithFakeContext( display, new Runnable() {
      public void run() {
        MySessionSingleton sessionSingleton = MySessionSingleton.getInstance();
        // singleton je nyni dostupny
      }
    } );
  }
};
new Thread( runnable ).start();
```

Výpis 3: Přístup k singletonu v Eclipse RAP [9]

3.3.2 Rozdíly RAP a RCP

Jak již bylo zmíněno, RAP nabízí pouze podmnožinu RCP API. To znamená, že v rámci RAP nefungují některé věci ohledně Workspace, JFace a některé události. RAP poskytuje takzvaný "sandbox", který umožňuje nahrát vývojářům jejich naimplementované věci, které nejsou k dispozici v rámci RAP. Toto uložiště je k dispozici ostatním vývojářům. Jako příklad jsou následně uvedeny některé části, které v Eclipse RAP stále chybí:

- GraphicalContext - komponenta, která umožňuje kreslení na obrazovku.
- StyledText - widget, umožňující zobrazovat text s možností nastavení vlastností písma.

4 Návrh aplikace

4.1 Navržení architektury MVC

Z důvodu snadné použitelnosti kódu na více platformách a ve více technologiích bylo zvoleno navržení aplikace pomocí architektury Model-View-Controller. Architektura je popsána v kapitole číslo 4.1.1. Tato architektura rozděluje implementaci kódu na tři nezávislé komponenty. Díky tomu při převodu aplikace na jinou technologii, stačí přepsat pouze komponentu, zajišťující zobrazení uživatelského rozhraní (například Swing, SWT atd.). Co patří do jednotlivých komponent ve vyvíjeném šachovém klientu lze vidět na obrázku číslo 7. Jak je v aplikaci tato architektura použita je vidět z třídního diagramu (Obrázek 8), který se týká části programu zajišťující zobrazování nabídek hráčů na serveru do tabulky. Následuje popis funkce jednotlivých komponent MVC v případě zobrazování nabídek na hru:

Model (SoughtInfo, Sought) - udržuje informace o nabízených hrách, jako například délka partie, uživatelské jméno soupeře atd.

View (SoughtTable, TableSorter) - zobrazuje tabulku s informacemi o nabídkách

Controller (SoughtController, SoughtParser) - reaguje na informace přicházející ze serveru včetně reakce na událost zvolení nabídky, pocházející od uživatele.

4.1.1 Popis architektury MVC

Model-View-Controller je softwarová architektura. Tato architektura odděluje doménový model aplikace od GUI. MVC bylo poprvé popsáno v roce 1979 vývojářem Trygve Reenskaug, který pracoval ve firmě Xerox na programovacím jazyku Smalltalk. MVC se skládá z následujících tří komponent:

Model (model) - jedná se doménový model aplikace, což znamená, že jsou zde reprezentována data, s nimiž aplikace pracuje.

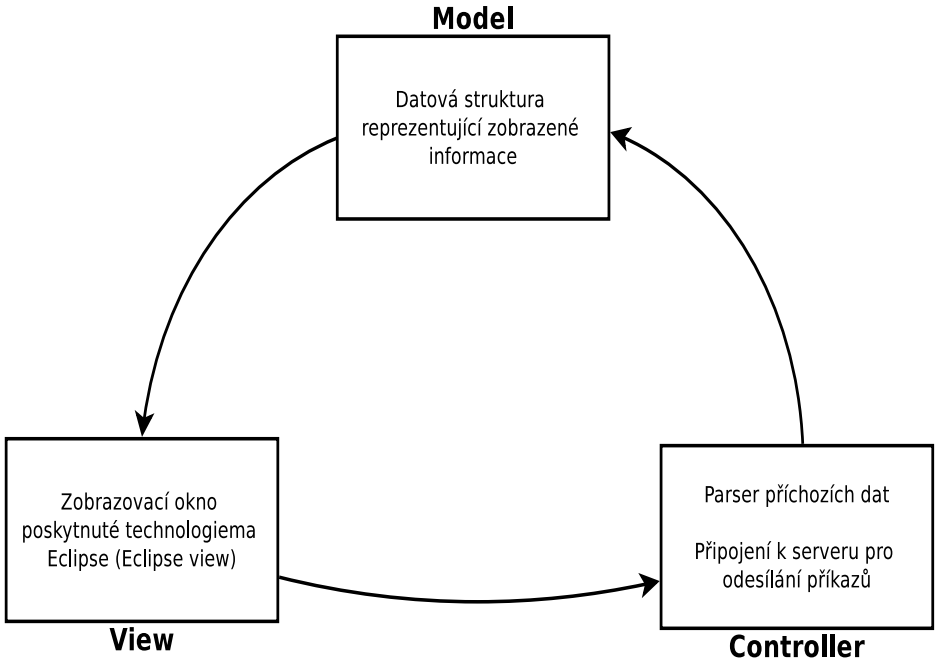
View (pohled) - tato část slouží k zobrazení informací z modelu uživateli.

Controller (řadič) - stará se o události vyvolané v komponentě view a zpracovává je například změnou dat v modelu. Jde o část, která stojí mezi modelem a pohledem a zajišťuje funkčnost systému.

Na obrázku číslo 9 lze vidět popis těchto částí včetně včetně interakcí, které mezi nimi probíhají.

4.2 Navržení připojení k serveru

K serveru se aplikace připojuje pomocí třídy Socket. Tato třída nám po připojení zpřístupní streamy pro příjem a odesílání informací. Jelikož je toto připojení pro celou aplikaci jedinečné, potřebujeme zajistit, aby k němu měla celá aplikace přístup. Toto je zajištěno



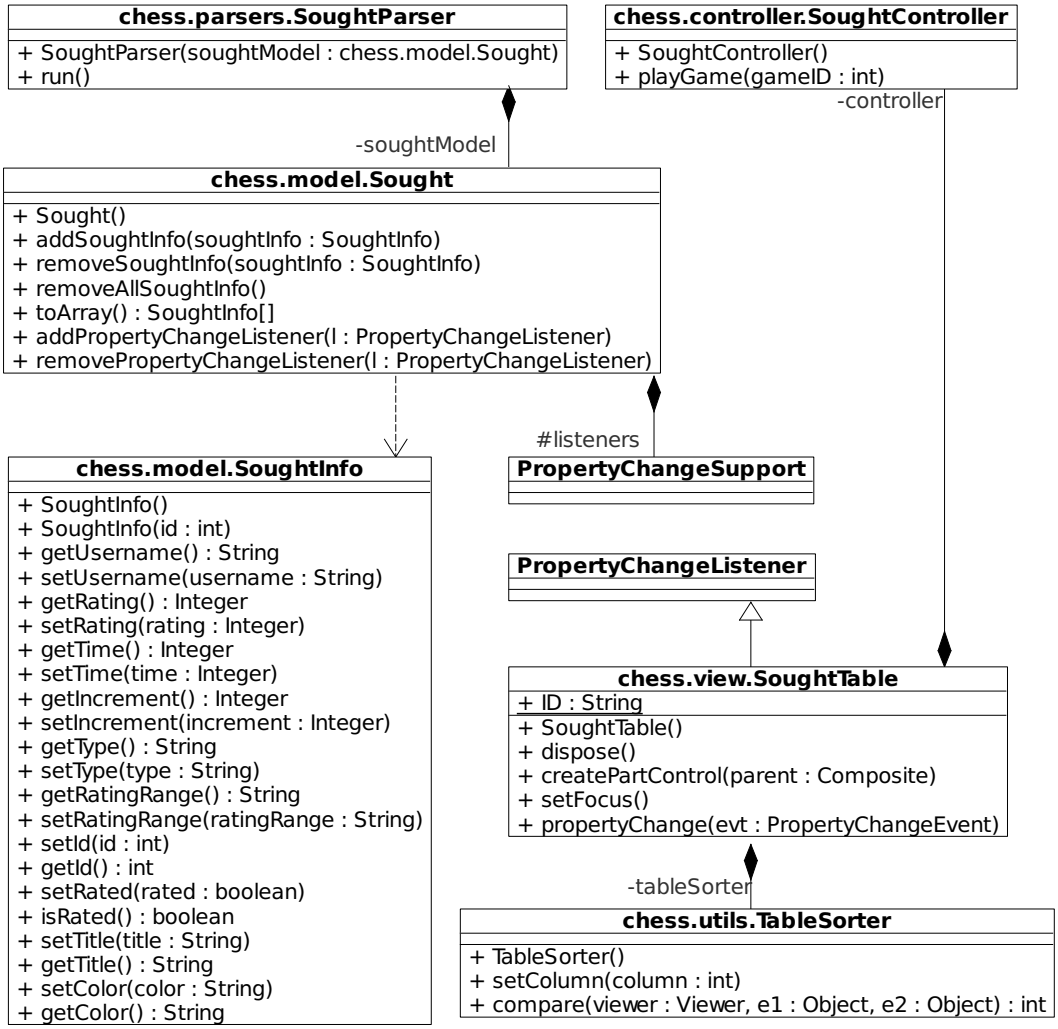
Obrázek 7: Ukázka jakým způsobem je využito MVC v aplikaci

pomocí návrhového vzoru Singleton, který je stručně popsán v kapitole číslo 4.2.1. Třídní diagram, popisující návrh připojení k serveru včetně zobrazování příchozích dat v konzoli, lze vidět na obrázku číslo 11.

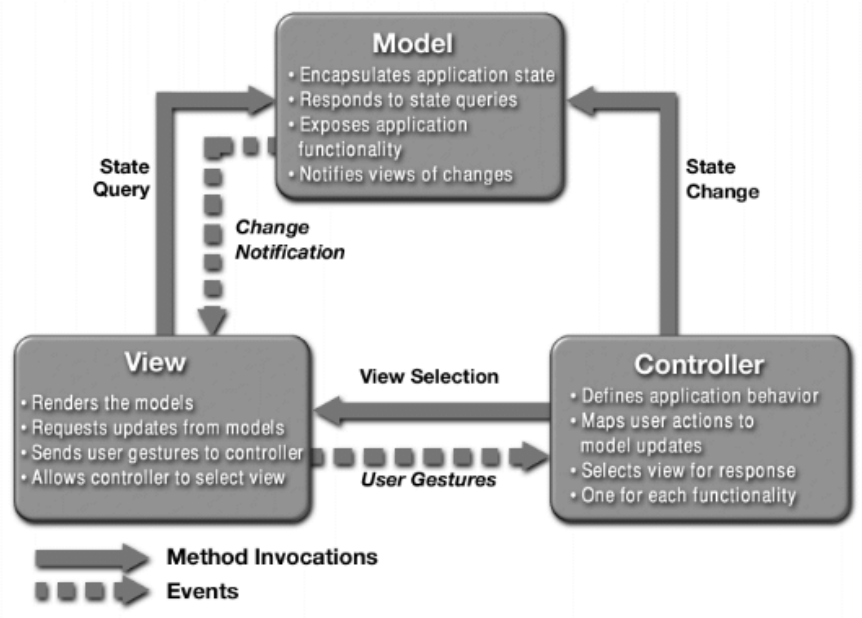
4.2.1 Návrhový vzor Singleton

Singleton patří do takzvaných tvořících návrhových vzorů (Creational patterns). Tento vzor zajišťuje, že daná třída bude mít jedinou instanci a poskytne k ní globální přístupový bod.[3] Této vlastnosti je využíváno v případě sdílení informací mezi objekty. Příklad implementace vzoru Singleton v jazyce Java lze vidět ve výpisu číslo 4 a popis pomocí třídního diagramu lze vidět na obrázku číslo 10.

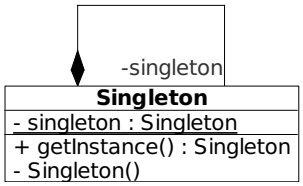
```
public class Singleton {

 //Vytvorime staticky  objekt
 private static  Singleton INSTANCE;

 //Vytvorime soukromy konstruktor
 private Singleton(){
 }

 //metoda pro vytvoreni instance singletonu
 public static  Singleton getInstance(){
      // jestlize  instance jeste  neexistuje  tak  vytvori  novou
```



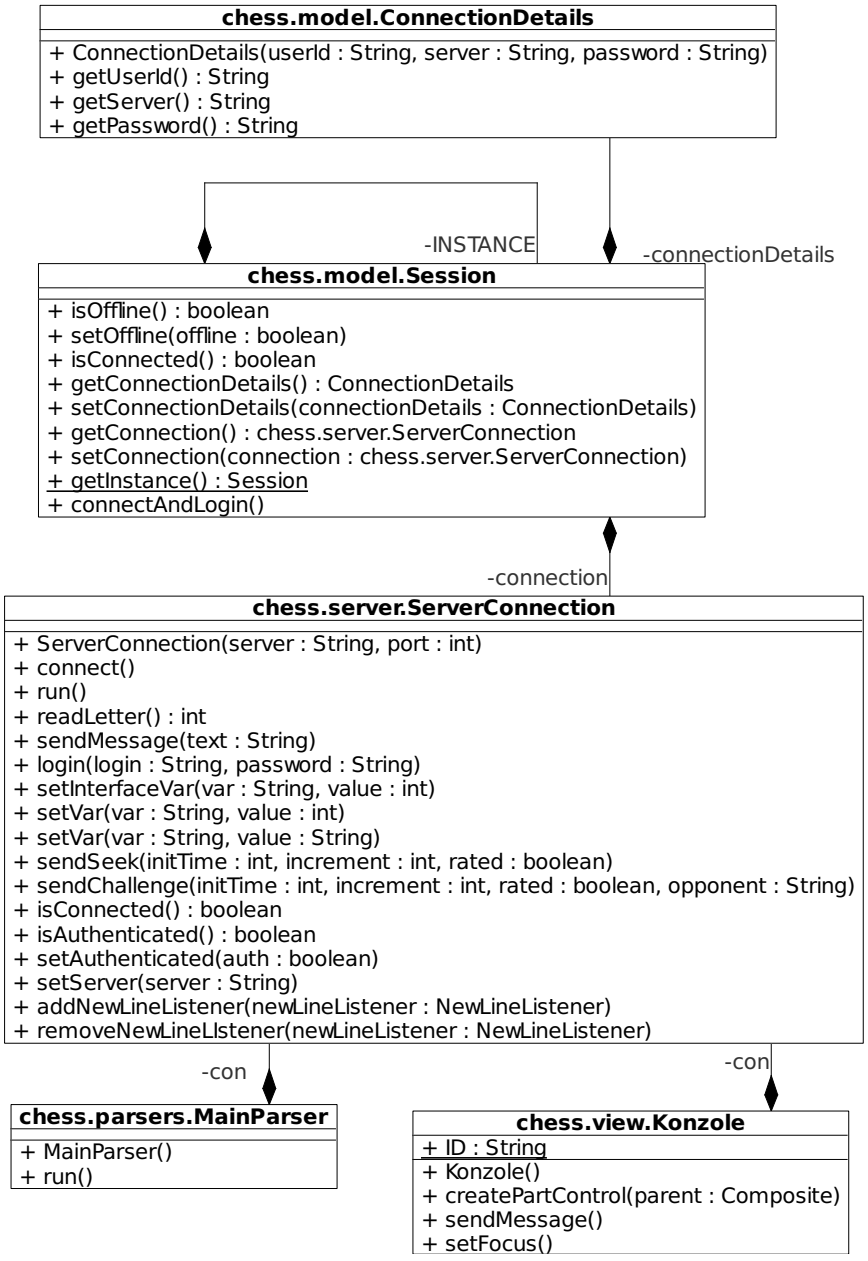
Obrázek 8: Třídní diagram - MVC návrh nabídek her



Obrázek 9: Architektura MVC [11]



Obrázek 10: Třídní diagram - Singleton



Obrázek 11: Ukázka využití vzoru Singleton v aplikaci

```
        if (INSTANCE == null) {
            INSTANCE = new Singleton();
        }
        return INSTANCE;
    }
```

Výpis 4: Implementace singletonu v jazyce Java

4.3 Parsovaní výstupu serveru

Pro rozeznávání jednotlivých informací zasílaných serverem je třeba tyto informace vhodně parsovat. Navrhl jsem parsery v aplikaci rozdělit do následujících kategorií. Každá kategorie se stará o určitý typ dat a je popsána v následujících podkapitolách:

4.3.1 MainParser

Jedná se o základní parser, který se stará pouze o to, že na začátku session umožní případné přihlášení uživatele účtem guest. Mezi hlavní úkoly parseru patří nastavení proměnných pro snadnější práci s informacemi ze serveru. Dále jsou uvedeny některé proměnné, které je třeba nastavit, včetně hodnoty proměnné a důvodu nastavení. K čemu proměnné slouží je popsáno v kapitole 2.2.1.

style 12 - jelikož je styl ve výchozím stavu nastaven na hodnotu 1 (Standard ICS board in ascii format), je potřeba nastavit styl na hodnotu 12 (style12), který lze jednoduše programově zpracovat.

unobserve 0 - je třeba nastavit tuto hodnotu, aby v případě pozorování ostatních partií bylo pozorování automaticky ukončeno v okamžiku začátku vlastní hry.

seek 0 - výchozí hodnota u guest účtu je 1. To znamená, že se zobrazují všechny informace o nabízených hrách. Tyto informace jsou pro účely aplikace již získávány pomocí proměnné seekinfo. Funkčnost programu neomezují, ale dělají nepřehledné přicházející informace v konzoli.

gin 0 - je potřeba zajistit, aby informace o hrách byly vypnuty, protože je používá DBParser pro získávání informací, které partie má pozorovat.

interface DoveChess 1.0 - nastaví informace o používaném klientu na aktuální název šachového klienta pro informaci ostatním uživatelům serveru.

autoflag 1 - pomáhá uživateli, protože automaticky kontroluje konec času na partii a v případě, že se tak stalo, ukončí hru.

seekinfo 1 - automaticky zobrazuje snadno zpracovatelné informace o nabízených partiích.

4.3.2 GameParser

Tento parser je určen pro zpracování informací o hře uživatele. Veškeré informace zobrazuje na šachovnici klienta včetně atributů hry, které jsou zobrazovány vedle šachovnice v infopanelu. Informace nutné ke zpracování hry uživatele:

Creating: garmm (1088) DoveCR (1087) rated blitz 5 0 - vytvoření hry včetně informací o hře.

{**Game 191 (garmm vs. DoveCR) garmm checkmated**} **0-1** - konec partie včetně výsledku a typu ukončení.

4.3.3 SoughtParser

Zajišťuje převod a zobrazování informací o nabízených hrách, které slouží uživateli pro výběr her s různými parametry. Parser zajišťuje obsluhu následujících událostí:

- Přidání nabídky - sleduje výskyt řádku začínajícího pomocí "<s>" a v případě výskytu použije informace z daného řádku pro zobrazení nabídky.
- Odebrání nabídky - sleduje výskyt řádku začínajícího pomocí "<sr>" a v případě výskytu odebere nabídku/nabídky definované číslem/čísly na daném řádku
- Zobrazení aktuálních nabídek ke hře - tuto událost zajistí nastavení proměnné seekinfo na hodnotu 1. Parser tedy hlídá konec partie a tuto událost vyvolá.
- Odebrání nabídek - hlídá událost odhlášení od serveru a v případě vzniku této události odebere veškeré informace o nabídkách, protože ty už nejsou dále platné. Stejná situace nastává také v případě vytvoření nové hry pro uživatele, jelikož nejsou tyto informace v době hraní partie potřebné.

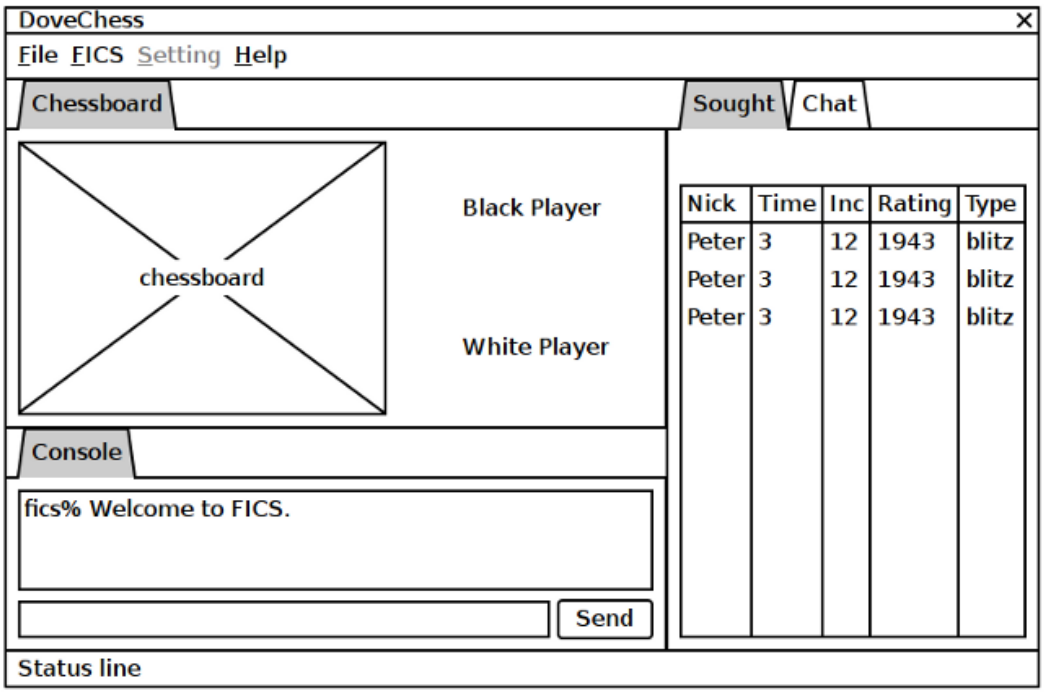
4.3.4 DBParser

Slouží k převodu informací z výstupu serveru do databáze aplikace. Díky nastavené proměnné gin je parser informován o začátku a konci každé partie na serveru. Pokouší se pozorovat každou hru (server FICS umožňuje pozorovat maximálně 30 her v jednom okamžiku) a ukládat její tahy do databáze včetně partií hraných uživatelem. Parser si udržuje seznam právě pozorovaných her, a to z důvodu konce hry, kdy může přiřadit výsledek partie jednotlivým tahům. Ukázka informací nutných ke zpracování, včetně popisu využití:

style12 - ukládá tahy k jednotlivým partiím

Game 74: OldRaptor (1890) abcgg (1987) rated standard 15 0 - informace o nové hře včetně atributů hry

{**Game 209 (GuestXBLR vs. GuestSZQH) Game aborted on move 1**} *
- informace o ukončené partii, včetně výsledku.



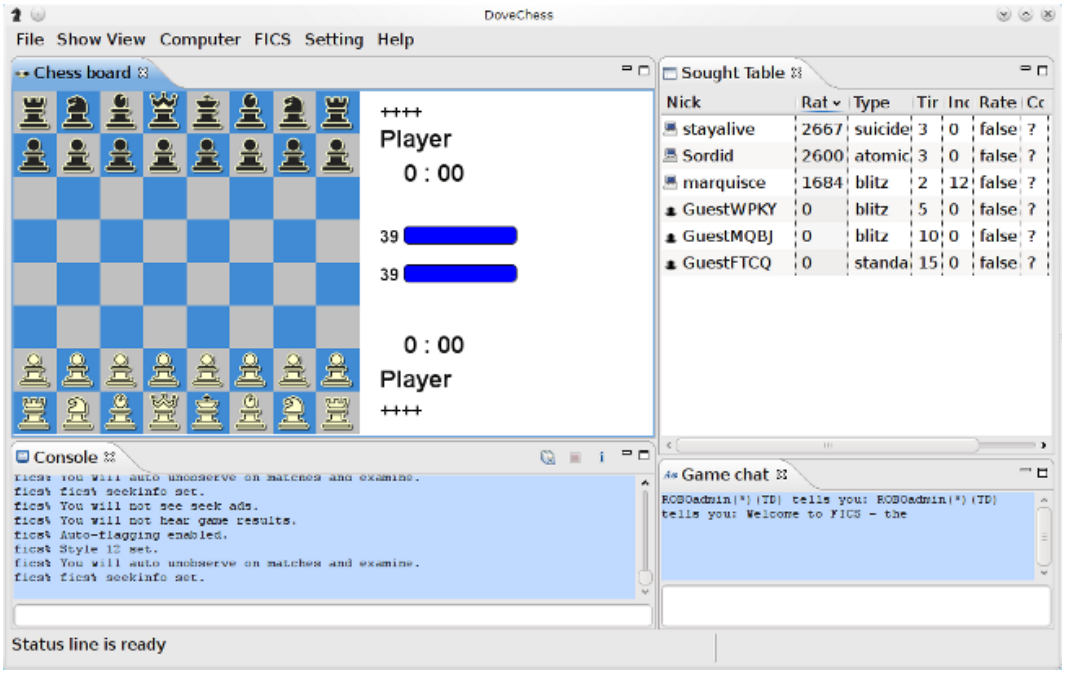
Obrázek 12: Návrh rozvržení GUI aplikace

4.4 Návrh rozhraní aplikace

Rozhraní je navrženo tak, aby byly při hře dostupné všechny důležité komponenty. Pro hru šachu jsou potřebné komponenty šachovnice, chatu a konzole. Zatímco při hledání hry a jiném využívání možností serveru jsou důležité komponenty konzole, tabulka nabízených her a menu. Na obrázku číslo 12 lze vidět, jak bylo uživatelské rozhraní navrženo a na obrázku číslo 13 je výsledný vzhled aplikace. Díky platformě Eclipse je však rozhraní velice flexibilní, takže si jej uživatel může nastavit podle svých potřeb. Aplikace je naimplementována tak, aby si toto nastavení uživatele zapamatovala i při ukončení aplikace a následném spuštění. Mezi další funkce pro pohodlnou práci uživatele s aplikací patří ikona umístěna v systémovém panelu. Tato ikona je vhodná při minimalizaci aplikace, protože program nezabírá žádné místo v aplikačním panelu. To je velmi praktické, protože aplikace může pracovat na pozadí (například při ukládání PGN souborů do databáze).

4.5 Problémy při realizaci

Při implementaci diplomové práce vznikaly určité problémy, kterým bych se chtěl v této kapitole krátce věnovat. Jednalo se o technologické problémy. Při realizaci vznikaly i další nepopsané komplikace, ale ty vycházely hlavně z neznalosti technologií.



Obrázek 13: Výsledný vzhled aplikace

4.5.1 Canvas Eclipse RAP

Největším problémem bylo zjištění, že Eclipse RAP nepodporuje komponentu Canvas. Tato komponenta umožňuje kreslení na obrazovku. V aplikaci je pomocí Canvas zobrazována celá šachovnice. Tedy nejdůležitější část programu. Tento problém lze vyřešit pomocí dvou možností, které jsou popsány v následujících podkapitolách. V aplikaci byla k vývoji z důvodu data vydání zvolena první možnost, popsána v kapitole 4.5.1.1.

4.5.1.1 GCCanvas od vývojáře Mirko Solazzi

Dne 16.1.2009 byla uvolněna první verze GCCanvas vývojářem Mirko Sollazi. Umožňuje veškeré kreslení na obrazovku, tak jak je tomu při vývoji RCP aplikace. Jediný nedostatek, který vývoj postihl byl ten, že nelze odebírat vložený text. Z toho důvodu musel být přepsán infopanel šachovnice za pomocí podporovaných komponent RWT (Label atd.).

4.5.1.2 Graphics Context

Graphics Context je již součástí Eclipse RAP a plně podporuje kreslení na obrazovku. Verze s podporou kreslení byla uvolněna 4.4.2010 ve verzi RAP 1.3 M7. Jedná se tedy pouze o vývojovou verzi RAP. Stabilní vydání Eclipse RAP 1.3 je plánováno na léto 2010.

4.5.2 Protokol FICS

Častým problémem, se kterým jsem se setkával, byla nejednoznačnost protokolu serveru FICS. Nebylo možno jednoduše rozeznat určité události vyvolané serverem. Jako příklad uvádím některé výstupy serveru a jejich význam:

'{Game' - takto začínající řádky se týkají jak oznámení o začátku partie tak, informace o konci. Je tedy složitější tyto události rozeznat.

'Game' - vedlejší informace týkající se hry. Jako atributy partie na začátku atd.

4.5.3 Další problémy

Mezi dalšími komplikacemi bych zmínil například nefunkční průhlednost u šachových figur Merida v operačních systémech Windows XP a Windows Vista. Nepodařilo se mi zjistit příčinu. Problém jsem vyřešil nahrazením stávajících figur novou sadou. Dále jsem se setkával s problémy při implementaci webové části s chybějícími komponentami v technologii Eclipse RAP oproti Eclipse RCP. Například se jedná o AboutDialog, který obsahuje informace o aplikaci. Problém byl vyřešen pomocí MessageDialogu, který již obě technologie obsahují.

5 Znalostní báze aplikace

V aplikaci slouží znalostní báze k ukládání tahů a k jejich následnému využití při nápovědě hráči, který má tuto pomoc zapnutou. Aby nedocházelo k podvodům, kdy si hráč nechá pomáhat počítačem proti reálným protihráčům, lze si na serveru FICS zaregistrovat počítačový účet. Lze také využít účet guest (návštěvník), kde je hraní s pomocí počítače povoleno.

5.1 Technologie

K vytvoření databáze byla použita embedded verze Java DB. Toto SŘBD je nyní vyvíjeno firmou Oracle (dříve firmou Sun) a vychází ze SŘBD Apache Derby a je distribuováno jako open source. Java DB je obsažena v balíku Java SE Development Kit.

5.1.1 Popis

Charakteristické prvky této databáze jsou:

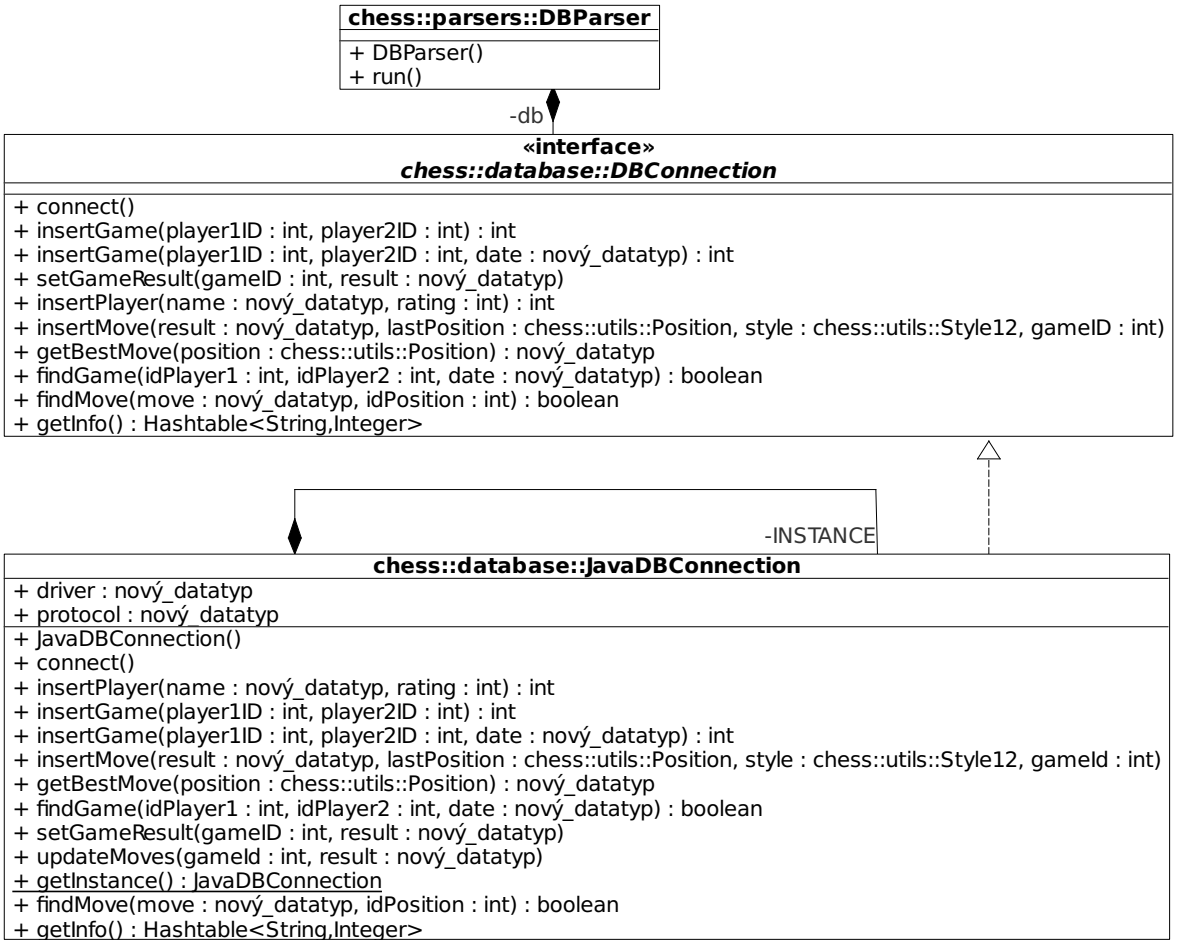
- jedná se o SQL databázi
- kompletně napsána v Javě
- jednoduše portovatelná
- serverová i embedded verze
- použitelnost na všech platformách, kde je nainstalována platforma Java SE
- velikost embedded verze pouze 2,5 MB
- volně dostupná s použitím Apache licence

5.1.2 Výhody

Důvody k použití embedded verze byly například to, že není nutné instalovat žádný server nebo mít nějaký k dispozici. Celá databáze se vytvoří v kořenové složce projektu (pokud není nadefinováno jinak). Jedna z dalších výhod je snadná přenositelnost celé databáze. Stačí pouze přenášet složku, ve které je umístěna. Další výhoda této databáze je, že je kompletně napsána v Javě. Tuto databázi lze tedy snadno připojit k jakémukoliv programu napsaném v Javě pomocí příslušných JAR souborů. V případě Eclipse již existují připravené pluginy, které se pouze naimportují do příslušného projektu.

5.1.3 Nevýhody

Mezi nevýhody bych zařadil absenci datového typu boolean. Tuto záležitost lze obejít pomocí datového typu smallint, který je v Java DB obsažen, ale z důvodu jednoduchosti se lépe pracuje s typem boolean, který je v jazyce Java obsažen. Další nevýhodou



Obrázek 14: Třídní diagram přístup aplikace ke znalostní bázi

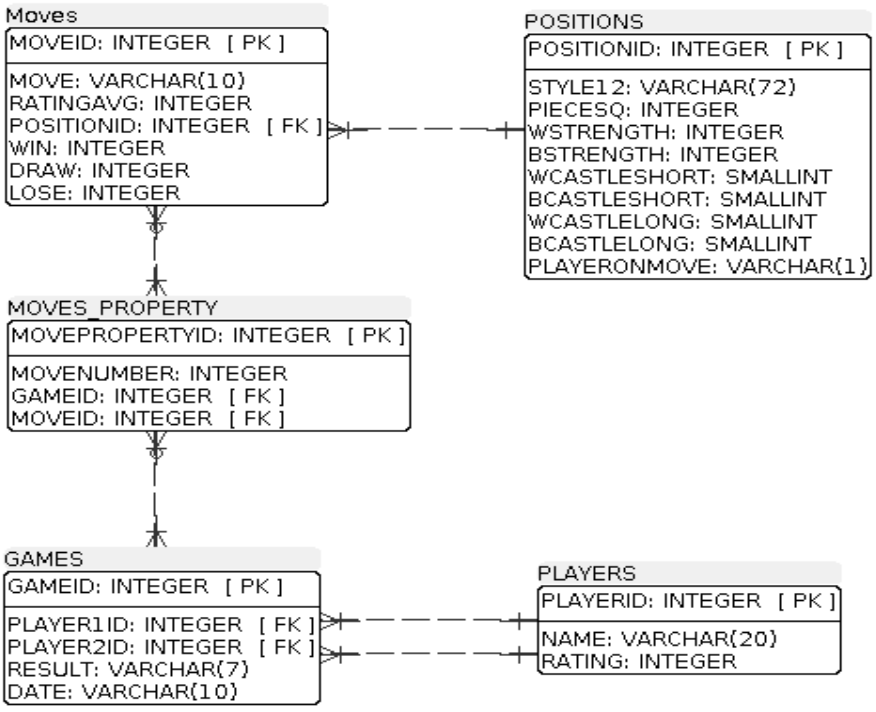
je například absence příkazu if v SQL syntaxi. Tento problém lze lehce vyřešit v programové části, ale za cenu delšího času zpracování, jelikož je potřeba více dotazů nad databází, které mají největší časovou náročnost.

5.2 Návrh databáze

Databáze je navržena tak, aby ukládala všechny pozice a tahy včetně všech informací potřebných pro další použití. Ukládání pozic je vytvořeno tak, aby v nich mohlo být rychle vyhledáváno, protože je třeba mít tyto tahy během hry rychle k dispozici. Programová část znalostní báze je navržena tak, aby bylo možno použít i jiné SŘBD pro ukládání partií. Tento návrh lze vidět z třídního diagramu na obrázku číslo 14.

5.2.1 ER diagram

Na obrázku číslo 15 lze vidět ER diagram navržené databáze.



Obrázek 15: ER diagram databáze

Moves

Atribut	Dat. typ	Vel.	Klíč	Null	Index	Popis
MOVEID	INTEGER	4	PK	N	Y	auto_increment
MOVE	VARCHAR	10	N	N	Y	tah
RATINGAVG	INTEGER	4	N	Y	N	prumerne hodnoceni
POSITIONID	INTEGER	4	FK	N	Y	id pozice
WIN	INTEGER	4	N	N	N	vyhrané
DRAW	INTEGER	4	N	N	N	remizované
LOSE	INTEGER	4	N	N	N	prohrané

Tabulka 1: Datový slovník - tabulka Moves

Moves_Property

Atribut	Dat. typ	Vel.	Klíč	Null	Index	Popis
MOVEPROPERTYID	INTEGER	4	PK	N	Y	auto_increment
MOVENUMBER	INTEGER	4	N	N	N	cislo tahu
GAMEID	INTEGER	4	N	N	Y	ke ktere hre patri
MOVEID	INTEGER	4	N	N	Y	nalezici tah

Tabulka 2: Datový slovník - tabulka Moves property

5.2.2 Význam jednotlivých tabulek

Players slouží k ukládání informací o všech hráčích, kteří hráli nějakou partii.

Games tabulka pro informace o jednotlivých hrách. Ukládá se dvojice hráčů a výsledek partie.

Moves zde se ukládají tahy z určitých pozic, na které je v vždy vytvořen odkaz.

Moves_Property ukládají se zde informace o tazích, zároveň se jedná o propojovací tabulku mezi tabulkou Games a Moves.

Positions nejdůležitější část, ve které se ukládají pozice. Jedná se o pozici figur na šachovnici včetně potřebných údajů, jako jsou informace o možnosti provést rošádu.

5.2.3 Datový slovník

Datový slovník databáze se skládá z tabulek číslo 1,2, 3, 4 a 5.

5.3 Popis vybraných atributů

MOVE - zapsaný tah k dané pozici. Tah je uložen ve formátu figura / zdrojové pole - cílové pole. Například P/e2-e4 udává tah pěšcem z pole e2 na pole e4. Jestliže je typ figury napsán velkým písmenem, jedná se o tah bílého. Jestliže malým, jedná

Players

Atribut	Dat. typ	Vel.	Klíč	Null	Index	Popis
PLAYERID	INTEGER	4	PK	N	N	auto_increment
NAME	VARCHAR	20	N	N	N	přihlašovací jméno
RATING	INTEGER	4	N	Y	N	skóre hráče

Tabulka 3: Datový slovník - tabulka Players

Games

Atribut	Dat. typ	Vel.	Klíč	Null	Index	Popis
GAMEID	INTEGER	4	PK	N	N	auto_increment
PLAYER1ID	INTEGER	4	FK	N	N	bílý hráč
PLAYER2ID	INTEGER	4	FK	N	N	černý hráč
RESULT	VARCHAR	7	N	Y	N	výsledek partie
DATE	VARCHAR	10	N	Y	N	datum

Tabulka 4: Datový slovník - tabulka Games

Positions

Atribut	Dat. typ	Vel.	Klíč	Null	Index	Popis
POSITIONID	INTEGER	4	PK	N	Y	auto_increment
STYLE12	VARCHAR	72	N	N	Y	pozice
PIECESQ	INTEGER	4	N	N	N	počet figur
WSTRENGTH	INTEGER	4	N	N	N	síla bílých figur
WCASTLESHORT	SMALLINT	1	N	N	N	krátká rošáda
WCASTLELONG	SMALLINT	1	N	N	N	dlouhá rošáda
BSTRENGTH	INTEGER	4	N	N	N	síla černých figur
BCASTLESHORT	SMALLINT	1	N	N	N	krátká rošáda
BCASTLELONG	SMALLINT	1	N	N	N	dlouhá rošáda
PLAYERONMOVE	VARCHAR	1	N	N	N	hráč na tahu

Tabulka 5: Datový slovník - tabulka Positons

se o tah černého. Rošáda se zapisuje speciální sekvencí pomocí sekvence O-O-O v případě dlouhé rošády a O-O v případě krátké rošády.

RATINGAVG - Jedná se o průměr hodnocení hráčů, kteří zahráli tah v dané pozici. Jestliže hráč nemá hodnocení (rating 0), tak se toto hodnocení do průměru nezapočítává z důvodu zkreslení průměru v případě, že by tento tah hrálo více lidí bez hodnocení než ostatní tahy.

WIN,DRAW,LOSE - hodnoty udávají, jak dopadly partie, ve kterých byl zahrán tento tah.

RATING - hodnocení hráče na serveru FICS v započítávaných hrách. Hodnocení se může rovnat nule, a to v případě, že se jedná o hráče bez hodnocení. To může nastat v případě, že se jedná o návštěvníka, který hraje nehodnocené partie. V takovém případě se hráč ukládá univerzálně se jménem "PLAYER_WITHOUT_RATING". U hráče se ukládá vždy aktuální hodnocení. To znamená hodnocení, které hráč měl při poslední uložené partii. Ukládá se zde jen jedno hodnocení ke všem typům her (lightning, blitz, standard). Je tomu tak z důvodu, že server již přihlíží k hodnocení hráčů při různých délkách partií. To znamená, že hráč hrající blitz partie a standard partie by měl mít vyšší hodnocení v případě standardních partií.

RESULT - jedná se o výsledek partie. Může nabývat čtyř hodnot:

- 1-0 výhra bílého hráče. Došlo k matu černého hráče nebo černému hráči došel čas určený ke hře.
- 0-1 výhra černého hráče. Došlo k matu bílého hráče nebo bílému hráči došel čas určený ke hře.
- 1/2-1/2 - remíza. Došlo k patové pozici nebo bylo uplatněno pravidlo o věčném šachu.
- null - V případě, že partie nebyla dohrána.

Na začátku partie se hodnota nastaví na null a při ohlášení konce hry se nastaví na příslušnou hodnotu výsledku.

STYLE12 - tento atribut slouží k ukládání pozice šachovnice. Každá pozice je v databázi unikátní (nesmí se opakovat). Je tomu tak proto, že i když každá pozice může nastat vícekrát, není třeba ji znovu ukládat, ale uložíme si pouze nový tah z dané pozice. Notace zde použitá je převzata z části notace style12, kterou používá server FICS pro zobrazení tahu. Jedná se o řetězec dlouhý 72 znaků, který obsahuje výpis jednotlivých řádků šachovnice oddělených mezerami. Reprezentace jednotlivých polí je následující:

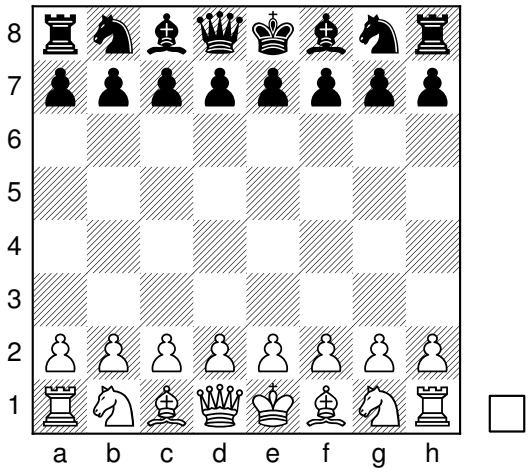
- K - Král (King)
- Q - Dáma (Queen)
- R - Věž (Rook)

- B - Střelec (Bishop)
- N - Kůň (Knight)
- P - Pěšec (Pawn)
- ”-” - pomlčkou je označeno pole na kterém neleží žádná figura

Stejně jako v případě tahů je velkým písmenem uváděna bíla figura a malým černá. Jako příklad je uvedena počáteční pozice šachovnice.

```
"rnbqkbnr pppppppp -------- -------- -------- --------
PPPPPPPP RNBQKBNR "
```

Na šachovnici můžete vidět jak je tento zápis reprezentován ve skutečnosti.



STRENGTH - zapisuje se síla figur příslušného hráče. Síla figur bílého je v databázi ukládána jako WSTRENGTH a síla figur černého jako BSTRENGTH. Síla figur znamená ohodnocení typů figur příslušnými hodnotami a sečtení hodnot v rámci jedné barvy. Hodnoty figur jsou dány následovně:

- Dáma = 9
- Věž = 5
- Střelec = 3
- Kůň = 3
- Pěšec = 1

Král se do síly nepočítá, protože jeho ztráta není možná. V některých publikacích je uváděna hodnota krále jako nekonečná.

CASTLE - ukládání informace o tom jestli je možno provést určitý typ rošády. V případě, že ano, je hodnota nastavena na 1, v případě, že ne, je hodnota nastavena na 0. Pro každou barvu hráče existují dva typy rošád a to dlouhá a krátká. V databázi jsou označovány jako SHORT a LONG. V počátku hry jsou všechny rošády nastaveny na hodnotu 1, protože jsou stále proveditelné. Ke znemožnění proveditelnosti rošády musí dojít tahem krále popřípadě příslušnou věží. Tudíž se informace neukládá o aktuálním stavu rošád, ale o tom, jestli v budoucích tazích bude možno příslušnou rošádu teoreticky provést.

5.4 Získávání dat

Aby program mohl nabízet pomocné tahy hráči, je potřeba mu nejprve nějaká data v podobě odehraných partií zprostředkovat. To lze zajistit třemi způsoby. Importováním již odehraných partií cizích lidí, sledováním aktuálně hraných cizích partií nebo vlastním hraním.

5.4.1 PGN import

PGN je univerzální formát pro zaznamenávání šachových partií. Soubor tohoto typu má příponu .pgn. Takovýto soubor může obsahovat jednu i více šachových partií. Jedná se o textový formát, který je považován za standart. Je navržen tak, aby byl snadno čitelný jak pro člověka, tak i pro počítač. Tyto soubory je možné získat například prostřednictvím serveru FICS. FICS udržuje všechny hry odehrané na tomto serveru ve své databázi a dává je k dispozici ve formátu PGN na adrese http://marcelk.net/logics/.

5.4.1.1 Popis formátu

Každá hra začíná hlavičkou, která obsahuje potřebné údaje netýkající se samotných tahů hráčů. Všechny údaje hlavičky jsou uzavřeny do hranatých závorek. Údaj má jasně definovaný název v anglickém jazyce. Hlavička obsahuje sedm povinných údajů:

Event: jméno turnaje nebo události.

Site: místo na kterém se partie konala. Zapisuje se ve formátu Město, Kraj, Stát. Stát je napsán pomocí tří písmenného kódu dané země.

Date: datum partie. Používá se formát RRRR.MM.DD. Může obsahovat "??" v případě že některý údaj není znám.

Round: číslo kola.

White: jméno hráče s bílými figurami. Zapisuje se ve formátu "příjmení, jméno".

Black: jméno hráče s černými figurami. Zapisuje se ve formátu "příjmení, jméno".

Result: výsledek partie. Může být použit znak "*" v případě, že hra stále probíhá.

Hlavička může obsahovat také volitelné údaje. Server FICS nabízí následující údaje:

TimeControl: údaj o počátečním čase ke hře.

ECO: typ zahájení partie.

Time: čas začátku partie.

WhiteElo: hodnocení bílého hráče.

BlackElo: hodnocení černého hráče.

Následuje textový zápis partie ve zkrácené notaci.

5.4.1.2 Ukázka hry zapsané pomocí formátu PGN [6]

```
[Event "F/S Return Match"]
[Site "Belgrade, Serbia JUG"]
[Date "1992.11.04"]
[Round "29"]
[White "Fischer, Robert J."]
[Black "Spassky, Boris V."]
[Result "1/2-1/2"]

1. e4 e5 2. Nf3 Nc6 3. Bb5 a6 4. Ba4 Nf6 5. O-O Be7 6. Re1 b5
7. Bb3 d6 8. c3 O-O 9. h3 Nb8 10. d4 Nbd7 11. c4 c6 12. cxb5
axb5 13. Nc3 Bb7 14. Bg5 b4 15. Nb1 h6 16. Bh4 c5 17. dxe5 Nxe4
18. Bxe7 Qxe7 19. exd6 Qf6 20. Nbd2 Nxd6 21. Nc4 Nxc4 22. Bxc4
Nb6 23. Ne5 Rae8 24. Bxf7+ Rxf7 25. Nxf7 Rxe1+ 26. Qxe1 Kxf7
27. Qe3 Qg5 28. Qxg5 hxg5 29. b3 Ke6 30. a3 Kd6 31. axb4 cxb4
32. Ra5 Nd5 33. f3 Bc8 34. Kf2 Bf5 35. Ra7 g6 36. Ra6+ Kc5
37. Ke1 Nf4 38. g3 Nxh3 39. Kd2 Kb5 40. Rd6 Kc5 41. Ra6 Nf2
42. g4 Bd3 43. Re6 1/2-1/2
```

5.4.2 Sledování hry

Na serveru FICS je umožněno sledovat až třicet právě probíhajících partií. Při sledování jsou tahy zobrazovány serverem pomocí protokolu style12, ze kterého lze snadno vyčíst informace o provedeném tahu. Výběr partií ke sledování je umožněn díky proměnné v_gin. Jedná se o proměnnou, která když je nastavena na true nebo 1, nastaví zobrazování informace o tom, že nějaká partie začíná nebo končí. Díky tomu lze sledovat celé hry. V aplikaci jsou nastavena kritéria, aby byly sledovány pouze hry typu standard a blitz. U kratších her již není zaručeno, že je tah dostatečně promyšlený a není zahrán jen proto, že je hráč v časové tísni.

5.4.3 Vlastní hra

Pokud hráč hraje partii a má zapnut počítačový režim aplikace, tak klient automaticky zaznamenává jednotlivé tahy partie, stejně jak je tomu u sledování her ostatních.

5.5 Ukládání partií

V každé partii jsou ukládány pozice do databáze včetně tahů, které z těchto pozic vychází. K uložení pozice lze tedy přistoupit až v momentě kdy je z ní proveden tah. Tato pozice je do databáze ukládána tak, že se vyhledá jestli daná pozice již v databázi existuje. V případě, že ne vytvoří se nový záznam o pozici a je k ní přiřazen daný tah. V případě, že ano je pouze upraven údaj o výsledku partie pro tuto pozici. Výsledek partie však v případě sledovaných partií nebo vlastních her není znám. I přesto je tah uložen a na konci hry je ke všem tahům patřících do této hry uložen výsledek. V případě, že není partie dohrána, zůstávají tyto tahy bez výsledku (pokud nebyly zahrány v předešlých hrách).

5.6 Výběr vhodného tahu

Výběr tahu slouží pro pomoc hráči při partii. Výběr takového tahu se skládá z několika fází. V následujícím výčtu jsou jednotlivé fáze popsány v řadě za sebou tak jak probíhají v programu.

- Výběr pozice - Tah je vybírán nejprve pomocí pozice. V tabulce Positions se vyhledá pozice se stejným rozložením všech figur. Dále jsou při výběru pozice brány v úvahu také jednotlivé rošády. Je tomu tak proto, že postavení figur může být shodné, ale v jednom z případů už není možno provést jednu z rošád, a to z důvodu již provedeným tahem králem, případně věží.
- Četnost tahu - Pro příslušnou pozici může existovat větší množství tahů. V tomto bodu jsou vybrány tahy, které jsou zahrány aspoň v 10% všech možných tahů.
- Rating tahu - zbylé tahy jsou seřazeny pomocí průměrného ratingu pro daný tah. Tento rating je vypočten průměrem ratingů všech hráčů, kteří tento tah zahráli.

5.7 Existující řešení

Pro vytváření znalostní báze k šachové a jí podobné hře již existují různé přístupy. V následující podkapitole bude popsán jeden z těchto přístupů.

5.7.1 GINA

Kenneth De Jong a Alan Schultz v roce 1988 zveřejnili jeden z přístupů [4] k řešení znalostní báze ve hře Othello (nazývaná také Reversi). Toto řešení pojmenovali GINA.

5.7.1.1 Architektura systému

Následuje popis architektury této znalostní báze ze tří pohledů: 1) jak je hra ve znalostní bázi reprezentována; 2) postup přidávání herních situací; 3) jak vybrat tah z báze.

Struktura znalostní báze

Jednou z hlavních části aplikace je báze, do které se ukládají pozorované partie. V této bázi se také vyhledávají tahy. Tato báze je složena z následujících částí:

orientovaný graf - obsahuje vrcholy, které zastupují jednotlivé herní situace. Tyto situace obsahují následující položky:

- rozložení pole hry
- hráč na tahu
- ukazatel na předchozí herní situaci
- ukazatele na následující herní situace
- frekvence - počet cest pod tímto vrcholem

Hrany v grafu reprezentují jednotlivé tahy, které se vyskytují mezi jednotlivými situacemi.

hašovací tabulka - zajišťuje rychlý přístup k jednotlivým vrcholům grafu. Jedná se o velice důležitou část, jelikož rychlost přístupu k jednotlivým vrcholům je ve velké bázi velice důležitá. Pro ještě větší rychlost je tato tabulka před používáním nahrána do paměti počítače.

Přidávání her do báze

Během hry jsou tahy jednotlivých hráčů ukládány do souboru. V okamžiku, kdy hra končí, jsou tahy zaznamenány do báze. Zaznamenávání začíná prvním tahem a na rozložení šachového pole včetně hráče na tahu se použije hašovací funkce. Výsledek této funkce se uloží do tabulky. Jestliže vrchol není v grafu nalezen, vytváří se nový vrchol. V určitých případech může nastat, že vrchol existuje, ale není v aktuální cestě průchodu. V takovémto případě se přidá ukazatel na tento vrchol jako na předchozí. V tuto chvíli, kdy již byly přidány všechny vrcholy a hrany pro hru, je třeba pro tyto tahy spočítat statistiky. Při výpočtu se vychází ze situace, popsané v listu cesty pro aktuální hru. Tyto informace se zpětně zapíší do jednotlivých vrcholů patřících do této hry. Tímto přístupem je zajištěno, že všechny vrcholy obsahují pouze statistiky vypočítané na základě odehraných kompletních her.

Využívání báze

Důležitou částí je použití uložených partií při vlastním hraní. Řešení GINA má implementován kvalitní program pro hraní Othello partií. Toho je využito v případě, že při hledání herní situace není žádná stejná situace nalezena, tehdy je použit program pro výpočet tahu. V případě existence vrcholu je v řešení GINA dán přednost tah z báze. Výběr tahu probíhá tak, že pokud existuje "dobrý" tah pro toto rozložení hracího pole (vede minimálně k remíze), je tento tah zvolen. V případě, že neexistuje takovýto tah, je vybrán takový, který má nejlepší poměr výsledků partií.

5.7.1.2 Posouzení účinnosti

Použití znalostní databáze v aplikaci pro hraní hry Othello se ukázalo jako efektivní. Autoři pozorovali GINA při hraní her a zaregistrovali zlepšení výkonnosti a výrazné snížení vytížení procesoru.

6 Závěr

6.1 Shrnutí

V této diplomové práci byl vyvinut multiplatformní šachový klient včetně desktopového i webového rozhraní dle zadání. Základem bylo prozkoumat technologie Eclipse RCP a RAP. Tyto technologie jsou velice obsáhlé a bylo třeba je pochopit pro snadnější implementaci aplikace. Dále byly prozkoumány služby serveru FICS, aby poté mohly být efektivně implementovány. Velký důraz byl kladen na návrh databáze pro ukládání tahů, aby byla umožněna rychlá nápověda při hraní šachů. Tato funkce byla naimplementována pouze do desktopové aplikace z důvodu, že v případě webové aplikace by ukládání tahů příliš zatěžovalo server.

6.2 Další vývoj

V budoucnu bych se chtěl věnovat hlavně rozšíření webové verze klienta pomocí Eclipse RAP. V první řadě bych chtěl přepsat komponentu, zajišťující vykreslování šachovnice pomocí nově vzniklého rozhraní, které Eclipse RAP nabízí. Poté mám zájem umístit webovou aplikaci na vlastní server a umožnit tak hraní těm uživatelům šachového serveru FICS, kterým nevyhovuje dosavadní rozhraní aplikace Jin. Dále bych chtěl oddělit znalostní bázi od RCP klienta z důvodu volného rozšíření klienta dalším uživatelům pod licencí GPL. Poté chci dále sledovat vývoj technologie Eclipse RAP a postupně implementovat další funkce, které budou vyvinuty.

David Juráň

7 Reference

[1] McAffer, Jeff and Lemieux, Jean-Michel, *Eclipse Rich Client Platform: Designing, Coding, and Packaging Java(TM) Applications*, New York: Addison-Wesley Professional, 2005. ISBN 0-321-33461-2.

[2] Lange, Fabian, *Eclipse Rich Ajax Platform: Bringing Rich Client to the Web*, New York: Apress Inc., 2008. ISBN 1-4302-1883-5.

[3] Gamma, Erich and Helm, Richard, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional; illustrated edition edition (November 10, 1994), ISBN 0-201-63361-2

[4] Kenneth A. De Jong and Alan C. Schultz, *Using Experience-Based Learning in Game Playing*, 1988.

[5] FICS, *FICS Help*, [online]. June 3, 1997 [cit. 2010-05-05]. FICS Help: style12. Dostupné z WWW: <http://www.freechess.org/Help/HelpFiles/style12.html>.

[6] Portable Game Notation In Wikipedia : *the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 6.8. 2005, 6.3. 2010 [cit. 2010-04-28]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Portable_Game_Notation>.

[7] *Eclipse documentation*, [online]. 2008 [cit. 2010-04-21]. RWT Theming. Dostupné z WWW: <http://help.eclipse.org/galileo/index.jsp>

[8] *Eclipse Rich Ajax Platform Home*, [online]. 2010 [cit. 2010-04-22]. Demos. Dostupné z WWW: <http://www.eclipse.org/rap/demos.php>

[9] *RAP wiki home*, [online]. 2010 [cit. 2010-04-26]. RAP/FAQ. Dostupné z WWW: <http://wiki.eclipse.org/RAP/FAQ>.

[10] Russel, Mark, *RAP vs GWT Which AJAX Technology is for you?*, [online]. 2008 [cit. 2010-04-15]. Dostupné z WWW: <http://www.slideshare.net/fireresq/rap-vs-gwt-which-ajax-technology-is-for-you-presentation>.

[11] Eckestein, Robert Java SE Application Design With MVC. *Feature Stories About Java Technology* [online]. March 2007, [cit. 2010-04-15]. Dostupný z WWW: <http://java.sun.com/developer/technicalArticles/javase/mvc/>.

8 Slovník pojmů

widget - znovupoužitelná část GUI.

framework - softwarová struktura, sloužící jako podpora při programování. Může obsahovat podpůrné knihovny, API a doporučený postup při vývoji.

session - trvající síťové spojení mezi klientem a serverem, zahrnující výměnu většího množství paketů.

sandbox - schránka obsahující aplikace a komponenty třetích stran.

konzole - jedná se o speciální příkazový řádek, který umožní posílat příkazy na server a zobrazovat jeho výstupy.

toolkit - nástroje.

embedded databáze - databázový systém, který je zabudován přímo do aplikace

parsování - syntaktická analýza. Proces transformace vstupního textu do datových struktur.

guest účet - účet sloužící nezaregistrovaným hráčům. Je určen pouze k hraní nehodnocených partií.

login hráče - jedinečné uživatelské jméno.

open source - jedná se o software s otevřeným zdrojovým kódem.

A Nasazení a spuštění aplikace

Webová aplikace

Pro nasazení aplikace je potřeba mít nainstalovaný a spuštěný server Apache Tomcat. Jednou z možností nasazení war souboru (soubor s příponou .war) na tento server je použití Tomcat webové aplikace ”manager webapp”, která bývá klasicky dostupná na adrese http://localhost:8080/manager/html. Zde je nutno v poli ”War file to deploy” pomocí tlačítka ”Choose File” vybrat soubor chess.war z přiloženého CD a použít tlačítko ”Deploy”. Pokud se vše podaří a je použito výchozí nastavení serveru tak by aplikace měla být dostupná na adrese: http://localhost:8080/chess/doveChess.

Desktopová aplikace

V případě desktopové aplikace je potřeba pouze vybrat aplikaci podle typu operačního systému a architektury procesoru. Aplikace jsou podle těchto vlastností rozdělené do složek. Uvnitř každé složky je spouštěcí soubor, kterým lze aplikaci spustit.

Příklad V případě že vlastníte operační systém Linux a procesor typu x86_84 spustíte aplikaci pomocí spouštěcího souboru umístěného ve složce linux.gtk.x86_64 .

B Ovládání znalostní báze

Pro používání znalostní báze je třeba být přihlášen v počítačovém režimu. Do tohoto režimu se lze dostat, když při přihlašování zaškrtnete volbu "Use computer mode".

Vkládání tahů

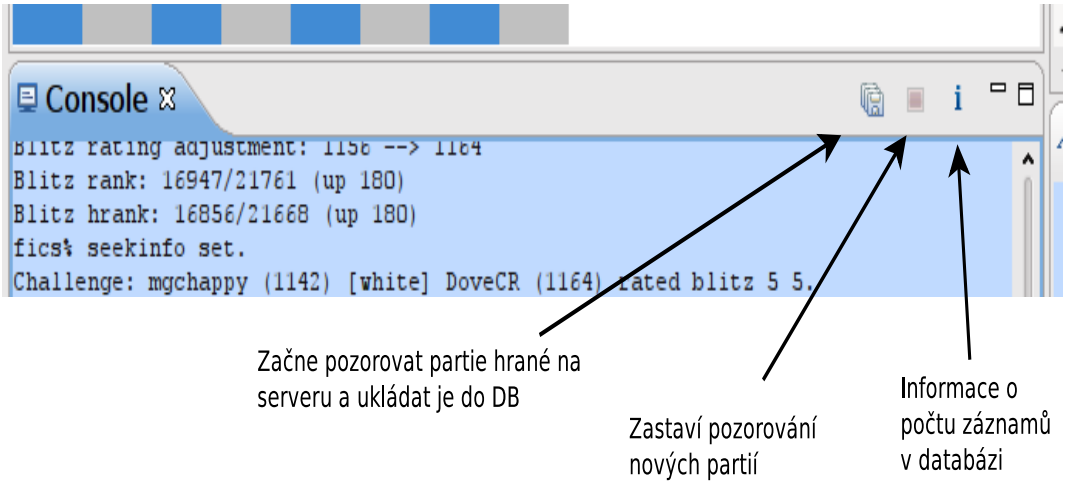
- Jedna z možností vložení tahů je pomocí PGN souborů. Takovýto soubor je možno vložit do databáze pomocí menu Computer− >Parse PGN.
- Další možnost je automatické ukládání při pozorování her. Pozorování her lze spustit pomocí příslušného tlačítka, které lze vidět na obrázku 16.

Zobrazování nabízených tahů

Možnost zobrazování nabízených tahů není ve výchozím stavu nastavena. Je proto třeba pomocí menu Setting− >Preferences otevřít dialog nastavení. V tomto dialogu v záložce Computer stačí zatrhnout možnost "Show moves for help" a potvrdit pomocí tlačítka Apply a OK.

Další funkce

- Ve zmiňovaném dialogu nastavení (Setting− >Preferences) lze také nastavit umístění a jméno databáze.
- Informace o počtu her, tahů a pozic uložených v databázi lze zjistit pomocí tlačítka "i" v okně konzole, které lze vidět na obrázku 16.



Obrázek 16: Ovládání znalostní báze